



## **Trident Protect**を使用してデータを保護する NetApp public and hybrid cloud solutions

NetApp  
February 04, 2026

# 目次

Trident Protectを使用してデータを保護する	1
Trident Protectを使用したOpenShift Container Platformのコンテナアプリのデータ保護	1
Trident Protect を使用して Red Hat OpenShift Virtualization の VM を保護する	14
オブジェクトストレージ用のApp Vaultを作成する	15
OpenShift Virtualization で VM を作成する	17
アプリを作成する	21
バックアップを作成してアプリを保護する	22
バックアップからの復元	25
スナップショットを使用してアプリを保護する	29
スナップショットからの復元	32
特定のVMを復元する	36
ビデオデモンストレーション	41

# Trident Protectを使用してデータを保護する

## Trident Protectを使用したOpenShift Container Platformのコンテナアプリのデータ保護

リファレンス ドキュメントのこのセクションでは、 Trident Protect を使用してコンテナアプリのスナップショットとバックアップを作成する方法について詳しく説明します。 NetApp Trident Protect は、 NetApp ONTAPストレージシステムとNetApp Trident CSIストレージ プロビジョナーによってサポートされるステートフル Kubernetes アプリケーションの機能と可用性を強化する高度なアプリケーション データ管理機能を提供します。 Trident Protect はアプリケーションのスナップショットとバックアップを作成します。つまり、永続ボリューム内のアプリケーション データのスナップショットとバックアップが作成されるだけでなく、アプリケーション メタデータのスナップショットとバックアップも作成されます。 Trident Protect によって作成されたスナップショットとバックアップは、次のいずれかのオブジェクト ストレージに保存し、後でそこから復元できます。

- AWS S3
- Azure BLOB ストレージ
- Google Cloud Storage
- オンタップS3
- ストレージグリッド
- その他のS3互換ストレージ

Trident Protect は、ロールベースのアクセス制御 (RBAC) の Kubernetes モデルを使用します。デフォルトでは、 Trident protect は trident-protect と呼ばれる単一のシステム名前空間とそれに関連付けられたデフォルトのサービス アカウントを提供します。組織内に多数のユーザーや特定のセキュリティ ニーズがある場合は、 Trident Protect の RBAC 機能を使用して、リソースや名前空間へのアクセスをより細かく制御できます。

Trident ProtectのRBACに関する追加情報は、 "[Tridentプロテクトのドキュメント](#)"



クラスター管理者は、デフォルトの trident-protect 名前空間内のリソースにアクセスでき、他のすべての名前空間内のリソースにもアクセスできます。ユーザーは、trident-protect 名前空間にスナップショットやバックアップ CR などのアプリケーション データ管理カスタム リソース (CR) を作成することはできません。ベストプラクティスとして、ユーザーはアプリケーション 名前空間にこれらの CR を作成する必要があります。

Trident Protectは、ドキュメントに記載されている手順に従ってインストールできます。 "[ここをクリックしてください](#)。" このセクションでは、コンテナ アプリケーションのデータ保護と Trident Protect を使用したアプリケーションの復元のワークフローについて説明します。 1.スナップショットの作成 (オンデマンドまたはスケジュール設定) 2.スナップショットからの復元 (同じ名前空間と異なる名前空間への復元) 3.バックアップ作成4.バックアップからの復元

## 前提条件

アプリケーションのスナップショットとバックアップを作成する前に、スナップショットとバックアップを保存するために、Trident Protect でオブジェクトストレージを構成する必要があります。これはバケット CR を使用して行われます。バケット CR を作成して構成できるのは管理者のみです。バケット CR は、Trident Protect では AppVault と呼ばれます。AppVault オブジェクトは、ストレージ バケットの宣言型 Kubernetes ワークフロー表現です。AppVault CR には、バックアップ、スナップショット、復元操作、SnapMirror レプリケーションなどの保護操作でバケットを使用するために必要な構成が含まれています。

この例では、ONTAP S3 をオブジェクトストレージとして使用する方法を示します。ONTAP S3 用の AppVault CR を作成するワークフローは次のとおりです。1. ONTAP クラスタの SVM に S3 オブジェクトストア サーバを作成します。2. オブジェクトストア サーバにバケットを作成します。3. SVM に S3 ユーザーを作成します。アクセスキーとシークレットキーを安全な場所に保管してください。4. OpenShift で、ONTAP S3 認証情報を保存するシークレットを作成します。5. ONTAP S3 用の AppVault オブジェクトを作成する

- ONTAP S3 用に Trident Protect AppVault を構成する\*\*
  - ONTAP S3 を AppVault として使用して Trident Protect を構成するためのサンプル yaml ファイル\*\*\*

```
# alias tp='tridentctl-protect'

appvault-secret.yaml

apiVersion: v1
stringData:
  accessKeyID: "<access key id created for a user to access ONTAP S3
bucket>"
  secretAccessKey: "corresponding Secret Access Key"
#data:
# base 64 encoded values
# accessKeyID: <base64 access key id created for a user to access
ONTAP S3 bucket>
# secretAccessKey: <base 64 Secret Access Key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque

appvault.yaml

apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
```

```

spec:
  providerConfig:
    azure:
      accountName: ""
      bucketName: ""
      endpoint: ""
    gcp:
      bucketName: ""
      projectID: ""
    s3:
      bucketName: <bucket-name for storing the snapshots and backups>
      endpoint: <endpoint IP for S3>
      secure: "false"
      skipCertValidation: "true"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-secret
  providerType: Ontaps3

# oc create -f appvault-secret.yaml -n trident-protect
# oc create -f appvault.yaml -n trident-protect

```

```

[root@localhost openshift-benchmark]#
[root@localhost openshift-benchmark]# tp get appvault -n trident-protect
+-----+-----+-----+-----+
| NAME | PROVIDER | STATE | AGE | ERROR |
+-----+-----+-----+-----+
| ontap-s3-appvault | OntapS3 | Available | 2d2h |       |
+-----+-----+-----+-----+
[root@localhost openshift-benchmark]# ■

```

postgresql アプリをインストールするためのサンプル yaml ファイル

```

postgres.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: postgres
spec:
  replicas: 1

```

```

selector:
  matchLabels:
    app: postgres
template:
  metadata:
    labels:
      app: postgres
spec:
  containers:
    - name: postgres
      image: postgres:14
      env:
        - name: POSTGRES_USER
          #value: "myuser"
          value: "admin"
        - name: POSTGRES_PASSWORD
          #value: "mypassword"
          value: "adminpass"
        - name: POSTGRES_DB
          value: "mydb"
        - name: PGDATA
          value: "/var/lib/postgresql/data/pgdata"
  ports:
    - containerPort: 5432
  volumeMounts:
    - name: postgres-storage
      mountPath: /var/lib/postgresql/data
  volumes:
    - name: postgres-storage
      persistentVolumeClaim:
        claimName: postgres-pvc
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgres-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
---
apiVersion: v1
kind: Service
metadata:

```

```
name: postgres
spec:
  selector:
    app: postgres
  ports:
    - protocol: TCP
      port: 5432
      targetPort: 5432
  type: ClusterIP
```

Now create the Trident protect application CR for the postgres app. Include the objects in the namespace postgres and create it in the postgres namespace.

```
# tp create app postgres-app --namespaces postgres -n postgres
```

```
[root@localhost RedHat]# tp get app -n postgres
+-----+-----+-----+-----+
| NAME | NAMESPACES | STATE | AGE |
+-----+-----+-----+-----+
| postgres-app | postgres | Ready | 24s |
+-----+-----+-----+-----+
[root@localhost RedHat]# ■
```

## スナップショットを作成する

### オンデマンド スナップショットの作成

```
# tp create snapshot postgres-snap1 --app postgres-app --appvault
ontap-s3-appvault -n postgres
Snapshot "postgres-snap1" created.
```

```
[root@localhost RedHat]# tp get snapshot -n postgres
```

NAME	APP REF	STATE	AGE	ERROR
postgres-snap1	postgres-app	Completed	19s	

```
[root@localhost RedHat]# ■
```

```
[root@localhost DataProtection]# oc get all,pvc,volumesnapshot -n postgres
apps.openshift.io/v1 DeploymentConfig is deprecated in v4.14+, unavailable in v4.1000+
kubevirt.io/v1 VirtualMachineInstancePresets is now deprecated and will be removed in v2.
NAME                                     READY   STATUS    RESTARTS   AGE
pod/postgres-cd9d6ccb-jfx49   1/1     Running   0          3h47m

NAME        TYPE      CLUSTER-IP       EXTERNAL-IP      PORT(S)      AGE
service/postgres  ClusterIP  172.30.132.112  <none>           5432/TCP   3h47m

NAME        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/postgres  1/1     1           1          3h47m

NAME        DESIRED  CURRENT    READY   AGE
replicaset.apps/postgres-cd9d6ccb  1        1         1      3h47m

NAME                           STATUS   VOLUME
ECLASS  VOLUMEATRIBUTESCLASS  AGE
persistentvolumeclaim/data-postgres-postgresql-0  Bound   pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  8Gi    RWO
ea-nas <unset>                4h40m
persistentvolumeclaim/postgres-pvc                 Bound   pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  5Gi    RWO
ea-nas <unset>                3h47m

NAME
READYVC
NAME SOURCEPVC          SOURCESNAPSHOTCONTENT  RESTORESIZE  SNAPSHOTCLASS      SNAPSHOTCONTENT
CREATIONTIME AGE
volumesnapshot.snapshot.storage.k8s.io/snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  true
postgres-pvc                                53676Ki  trident-snapshotclass  snapcontent-796ea7f8-59a0-493e-bbd8-3a
e76fe9036c  13m      13m
volumesnapshot.snapshot.storage.k8s.io/snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  true
data-postgres-postgresql-0                   368Ki   trident-snapshotclass  snapcontent-86a464d4-ffd8-4279-9cf7-88
88a097c001  13m      13m
```

スケジュールの作成 次のコマンドを使用すると、スナップショットは毎日 15:33 に作成され、2 つのスナップショットとバックアップが保持されます。

```
# tp create schedule schedule1 --app postgres-app --appvault ontap-s3-
appvault --backup-retention 2 --snapshot-retention 2 --granularity
Daily --hour 15 --minute 33 --data-mover Restic -n postgres
Schedule "schedule1" created.
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
+-----+-----+-----+-----+-----+-----+
| NAME | APP | SCHEDULE | ENABLED | STATE | AGE | ERROR |
+-----+-----+-----+-----+-----+-----+
| schedule1 | postgres-app | Daily:hour=15,min=33 | true | 17s | | |
+-----+-----+-----+-----+-----+-----+
[root@localhost DataProtection]# ■
```

### yamlを使用してスケジュールを作成する

```
# tp create schedule schedule2 --app postgres-app --appvault ontap-s3-
appvault --backup-retention 2 --snapshot-retention 2 --granularity
Daily --hour 15 --minute 33 --data-mover Restic -n postgres --dry-run >
hourly-snapshotschedule.yaml
```

```
cat hourly-snapshotschedule.yaml
```

```
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: schedule2
  namespace: postgres
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: postgres-app
  backupRetention: "2"
  dataMover: Restic
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  #hour: "15"
  minute: "33"
  recurrenceRule: ""
  snapshotRetention: "2"
status: {}
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
+-----+-----+-----+-----+-----+-----+
| NAME | APP | SCHEDULE | ENABLED | STATE | AGE | ERROR |
+-----+-----+-----+-----+-----+-----+
| schedule1 | postgres-app | Daily:hour=15,min=33 | true | 8d7h | | |
| schedule2 | postgres-app | Hourly:min=33 | true | 8d7h | | |
+-----+-----+-----+-----+-----+-----+
[root@localhost DataProtection]# ■
```

Ad  
Go

このスケジュールで作成されたスナップショットを表示できます。

```
[root@localhost DataProtection]# tp get snap -n postgres
+-----+-----+-----+-----+-----+
| NAME | APP REF | STATE | AGE | ERROR |
+-----+-----+-----+-----+-----+
| hourly-3f1ee-20250214183300 | postgres-app | Completed | 19s | |
| postgres-snap1 | postgres-app | Completed | 1h25m | |
+-----+-----+-----+-----+-----+
[root@localhost DataProtection]# ■
```

ボリュームのスナップショットも作成されます。

```
[root@localhost DataProtection]# oc get volumesnapshots -n postgres
NAME          CREATIONTIME   AGE    READYTOUSE   SOURCEPVC
snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  114m   true      postgres-pvc
snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  114m   true      data-postgres-postgresql-0
snapshot-ce75a274-ecb2-48c9-a0a5-94c10f8e6cb1-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  30m    true      postgres-pvc
snapshot-ce75a274-ecb2-48c9-a0a5-94c10f8e6cb1-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  30m    true      data-postgres-postgresql-0
[root@localhost DataProtection]# ■
```

アプリケーションを削除してアプリケーションの損失をシミュレートします

```
# oc delete deployment/postgres -n postgres
# oc get pod,pvc -n postgres
No resources found in postgres namespace.
```

スナップショットから同じ名前空間に復元する

```
# tp create sir postgres-sir --snapshot postgres/hourly-3flee-20250214183300 -n postgres
SnapshotInplaceRestore "postgres-sir" created.
```

NAME	APPVAULT	STATE	AGE	ERROR
postgres-sir	ontap-s3-appvault	Completed	2m39s	

アプリケーションとその PVC は同じ名前空間に復元されます。

```
[root@localhost DataProtection]# oc get pods,pvc -n postgres
NAME          READY   STATUS    RESTARTS   AGE
pod/postgres-cd9d6ccb-x85tg   1/1     Running   0          98s

NAME                           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/data-postgres-postgresql-0   Bound   pvc-b2cd67fd-fe4d-49b1-9e06-a5bf7be575e   8Gi        RWO          sc-zonea-nas   <unset>           105s
persistentvolumeclaim/postgres-pvc                  Bound   pvc-2d549395-0cc6-4529-b2b9-7361fb14fa8   5Gi        RWO          sc-zonea-nas   <unset>           105s
[root@localhost DataProtection]#
```

スナップショットから別の名前空間に復元する

```
# tp create snapshotrestore postgres-restore --snapshot postgres/hourly-3flee-20250214183300 --namespace-mapping postgres:postgres-restore -n postgres-restore
SnapshotRestore "postgres-restore" created.
```

NAME	APPVAULT	STATE	AGE	ERROR
postgres-restore	ontap-s3-appvault	Completed	1m15s	

アプリケーションが新しい名前空間に復元されたことがわかります。

```
[root@localhost DataProtection]#
[root@localhost DataProtection]# oc get pods,pvc -n postgres-restore
NAME          READY   STATUS    RESTARTS   AGE
pod/postgres-cd9d6ccb-pfxw9   1/1     Running   0          8d

NAME                           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
persistentvolumeclaim/data-postgres-postgresql-0   Bound   pvc-3dff4e42-828d-49f9-84e9-4daf75479292   8Gi        RWO          sc-zonea-nas
persistentvolumeclaim/postgres-pvc                  Bound   pvc-f57321b2-f21f-4eb7-8f86-4a17f915318b   5Gi        RWO          sc-zonea-nas
[root@localhost DataProtection]#
```

## バックアップを作成する

### オンデマンド バックアップの作成

```
# tp create backup postgres-backup1 --app postgres-app --appvault
ontap-s3-appvault -n postgres
Backup "postgres-backup1" created.
```

```
[root@localhost DataProtection]# tp get backup -n postgres
```

NAME	APP REF	STATE	AGE	ERROR
backup1	postgres-app	Completed	5d12h	
daily-feac1-20250222153300	postgres-app	Completed	1d10h	
daily-feac1-20250223153300	postgres-app	Completed	10h36m	
hourly-3f1ee-20250224003300	postgres-app	Completed	1h36m	
hourly-3f1ee-20250224013300	postgres-app	Completed	36m27s	
postgres-backup1	postgres-app	Completed	6m19s	

### バックアップのスケジュールを作成

上記のリストにある毎日のバックアップと毎時のバックアップは、以前に設定されたスケジュールから作成されます。

```
# tp create schedule schedule1 --app postgres-app --appvault ontap-s3-
appvault --backup-retention 2 --snapshot-retention 2 --granularity
Daily --hour 15 --minute 33 --data-mover Restic -n postgres
Schedule "schedule1" created.
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
```

NAME	APP	SCHEDULE	ENABLED	STATE	AGE	ERROR
schedule1	postgres-app	Daily:hour=15,min=33	true		9d8h	
schedule2	postgres-app	Hourly:min=33	true		9d8h	

## バックアップからの復元

データ損失をシミュレートするには、アプリケーションと PVC を削除します。

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME           READY   STATUS    RESTARTS   AGE
postgres-cd9d6ccb-dftkt  1/1     Running   0          11s
[root@localhost DataProtection]# oc get deployment -n postgres
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
postgres   1/1     1            1            20s
[root@localhost DataProtection]# oc delete deployment/postgres -n postgres
deployment.apps "postgres" deleted
[root@localhost DataProtection]# oc get pods -n postgres
No resources found in postgres namespace.
[root@localhost DataProtection]# oc get pvc -n postgres
NAME                           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTE
SCCLASS   AGE
data-postgres-postgresql-0   Bound    pvc-b2cd67fd-fe4d-49b1-9e06-a53bf7be575e   8Gi        RWO          sc-zonea-nas   <unset>
5d13h
postgres-pvc                 Bound    pvc-2d549395-0cc6-4529-b2b9-7361bfb14fa8   5Gi        RWO          sc-zonea-nas   <unset>
5d13h
[root@localhost DataProtection]# oc delete pvc/data-postgres-postgresql-0 -n postgres
persistentvolumeclaim "data-postgres-postgresql-0" deleted
[root@localhost DataProtection]# oc delete pvc/postgres-pvc -n postgres
persistentvolumeclaim "postgres-pvc" deleted
[root@localhost DataProtection]# oc get pvc -n postgres
No resources found in postgres namespace.
[root@localhost DataProtection]#
```

同じ名前空間に復元 #tp create bir postgres-bir --backup postgres/hourly-3f1ee-20250224023300 -n postgres BackupInplaceRestore "postgres-bir" が作成されました。

```
[root@localhost DataProtection]# tp get bir -n postgres
+-----+-----+-----+-----+
|      NAME      | APPVAULT      | STATE      | AGE      | ERROR      |
+-----+-----+-----+-----+
| postgres-bir | ontap-s3-appvault | Completed | 2m19s |           |
+-----+-----+-----+-----+
```

アプリケーションと PVC は同じ名前空間に復元されます。

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME           READY   STATUS    RESTARTS   AGE
postgres-cd9d6ccb-t857w  1/1     Running   0          10m
[root@localhost DataProtection]# oc get pvc -n postgres
NAME                           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTE
SCCLASS   AGE
data-postgres-postgresql-0   Bound    pvc-0a849c19-16fe-466f-9733-85e82a8b1677   8Gi        RWO          sc-zonea-nas   <unset>
10m
postgres-pvc                 Bound    pvc-ded304ea-02d4-4225-b606-63007666ad66   5Gi        RWO          sc-zonea-nas   <unset>
```

別の名前空間に復元 新しい名前空間を作成します。バックアップから新しい名前空間に復元します。

```
[root@localhost DataProtection]# oc create ns postgres-restore-from-backup
namespace/postgres-restore-from-backup created
[root@localhost DataProtection]# tp create backuprestore postgres-restore-from-backup --backup postgres/postgres-backup1 --namespace-map
ping postgres:postgres-restore-from-backup -n postgres-restore-from-backup
BackupRestore "postgres-restore-from-backup" created.
[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME          STATE    ERROR   AGE
postgres-restore-from-backup  Running   37s
[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME          STATE    ERROR   AGE
postgres-restore-from-backup  Running   56s
[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME          STATE    ERROR   AGE
postgres-restore-from-backup  Completed 2m52s
[root@localhost DataProtection]# oc get pods -n postgres-restore-from-backup
NAME            READY  STATUS    RESTARTS   AGE
postgres-cd9d6ccb-p659p  1/1    Running   0          2m9s
[root@localhost DataProtection]# oc get pvc -n postgres-restore-from-backup
NAME          STATUS    VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUT
ESCLASS   AGE
data-postgres-postgresql-0  Bound    pvc-36df7399-95da-4c67-a621-af9434015bdb  8Gi       RWO          sc-zonea-nas  <unset>
2m18s
postgres-pvc           Bound    pvc-633de3aa-a4f9-4f3b-93cc-e91afbd4fe02  5Gi       RWO          sc-zonea-nas  <unset>
2m18s
[root@localhost DataProtection]#
```

Activate Windows  
sc-zonea-nas <unset>  
Go to Settings to activate Windows.

## アプリケーションの移行

アプリケーションを別のクラスターに複製または移行するには(クラスター間複製を実行する)、ソースクラスターでバックアップを作成し、そのバックアップを別のクラスターに復元します。宛先クラスターにTrident Protectがインストールされていることを確認します。

ソースクラスターで、次の図に示す手順を実行します。

```
[root@localhost DataProtection]# tp create backup postgres-backup-cluster1 --app postgres-app --appvault ontap-s3-appvault -n postgres
Backup "postgres-backup-cluster1" created.
[root@localhost DataProtection]# tp get backup -n postgres
+-----+-----+-----+-----+
| NAME | APP REF | STATE | AGE   | ERROR |
+-----+-----+-----+-----+
| backup1 | postgres-app | Completed | 5d14h |      |
| daily-feac1-20250222153300 | postgres-app | Completed | 1d12h |      |
| daily-feac1-20250223153300 | postgres-app | Completed | 12h18m |      |
| hourly-3flee-20250224023300 | postgres-app | Completed | 1h18m |      |
| hourly-3flee-20250224033300 | postgres-app | Completed | 18m26s |      |
| postgres-backup-cluster1 | postgres-app | Running  | 35s  |      |
| postgres-backup1 | postgres-app | Completed | 1h48m |      |
+-----+-----+-----+-----+
[root@localhost DataProtection]# tp get backup -n postgres
+-----+-----+-----+-----+
| NAME | APP REF | STATE | AGE   | ERROR |
+-----+-----+-----+-----+
| backup1 | postgres-app | Completed | 5d14h |      |
| daily-feac1-20250222153300 | postgres-app | Completed | 1d12h |      |
| daily-feac1-20250223153300 | postgres-app | Completed | 12h19m |      |
| hourly-3flee-20250224023300 | postgres-app | Completed | 1h19m |      |
| hourly-3flee-20250224033300 | postgres-app | Completed | 19m41s |      |
| postgres-backup-cluster1 | postgres-app | Completed | 1m50s |      |
| postgres-backup1 | postgres-app | Completed | 1h49m |      |
+-----+-----+-----+-----+
```

Activate Windows  
Go to Settings to activate Windows.

ソースクラスターから、コンテキストを宛先クラスターに切り替えます。次に、宛先クラスター コンテキストから AppVault にアクセスできることを確認し、宛先クラスターから AppVault の内容を取得します。

```
[root@localhost DataProtection]# kubectl config use-context default/api-bm-cluster5-min-ocpv-sddc-netapp-com:6443/kube:admin
Switched to context "default/api-bm-cluster5-min-ocpv-sddc-netapp-com:6443/kube:admin".
[root@localhost DataProtection]# tp get appvault -n trident-protect
+-----+-----+-----+-----+
| NAME | PROVIDER | STATE | AGE   | ERROR |
+-----+-----+-----+-----+
| ontap-s3-appvault | OntapS3 | Available | 3d6h |      |
+-----+-----+-----+-----+
[root@localhost DataProtection]# tp get appvaultcontent ontap-s3-appvault --show-resources backup --show-paths
+-----+-----+-----+-----+
| CLUSTER | APP PATH | TYPE | NAME | TIMESTAMP |
+-----+-----+-----+-----+
| ocp-cluster11 | bbox | backup | bboxbackup1 | 2025-01-17 15:57:49 (UTC) | bbox_68b4f05f-d5f1-4304-8c69-ad67514fe393/backups/bboxbackup1_3960c945-8ee0-42fe-945c-c57bb0a3af6f |
| ocp-cluster11 | postgres-app | backup | backup1 | 2025-02-18 13:31:50 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/backup1_28e1bd9a-9b04-4412-8b96-811f9b62e2e3 |
| ocp-cluster11 | postgres-app | backup | daily-feac1-20250222153300 | 2025-02-22 15:34:44 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/daily-feac1-20250222153300_23d1386b-09f0-456f-aa49-a5865fd48abd |
| ocp-cluster11 | postgres-app | backup | daily-feac1-20250223153300 | 2025-02-23 15:34:42 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/daily-feac1-20250223153300_c492a4d1-38a9-4472-9684-4705c12a206d |
| ocp-cluster11 | postgres-app | backup | hourly-3flee-20250224033300 | 2025-02-24 03:34:44 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/hourly-3flee-20250224033300_3d09ab81-f6fe-47fa-a699-28006160cdbc |
| ocp-cluster11 | postgres-app | backup | hourly-3flee-20250224043300 | 2025-02-24 04:34:47 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/hourly-3flee-20250224043300_66805e4f-7631-48a6-98f7-d34bb8626031 |
| ocp-cluster11 | postgres-app | backup | postgres-backup-cluster1 | 2025-02-24 03:52:36 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/postgres-backup-cluster1_ec0ed3f3-5500-4e72-afa8-117a04a0b1c3 |
```

リストのバックアップパスを使用して、以下のコマンドに示すように、backuprestore CR オブジェクトを作成します。

```
# tp create backuprestore backup-restore-cluster2 --namespace-mapping
postgres:postgres --appvault ontap-s3-appvault --path postgres-
app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/postgres-backup-
cluster1_ec0ed3f3-5500-4e72-afa8-117a04a0b1c3 -n postgres
BackupRestore "backup-restore-cluster2" created.
```

```
[root@localhost DataProtection]# tp get backuprestore -n postgres
```

NAME	APPVAULT	STATE	AGE	ERROR
backup-restore-cluster2	ontap-s3-appvault	Completed	12m41s	

アプリケーション ポッドと PVC が宛先クラスターに作成されたことがわかります。

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME           READY   STATUS    RESTARTS   AGE
postgres-cd9d6ccb-2lvcq  1/1    Running   0          13m
[root@localhost DataProtection]# oc get pvc -n postgres
NAME           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUT
ESCLASS   AGE
data-postgres-postgresql-0   Bound    pvc-872a5182-601b-4848-b410-fef368337d07   8Gi        RWO          sc-zoneb-san <unset>
13m
postgres-pvc            Bound    pvc-caf9fa71-76a8-4645-9bb5-2ed72e72948b   5Gi        RWO          sc-zoneb-san <unset>
13m
[root@localhost DataProtection]#
```

## Trident Protect を使用して Red Hat OpenShift Virtualization の VM を保護する

スナップショットとバックアップを使用して、OpenShift Virtualization 内の VM を保護します。この手順には、ONTAP S3 オブジェクトストレージを使用して AppVault を作成し、Kubernetes リソースオブジェクト、永続ボリューム、内部イメージなどの VM データをキャプチャするように Trident Protect を構成し、必要に応じてデータを復元することが含まれます。

OpenShift Virtualization 環境の仮想マシンは、OpenShift Container プラットフォームのワーカーノードで実行されるコンテナー化されたアプリケーションです。VM のメタデータと永続ディスクを保護して、それらが失われたり破損したりした場合に回復できるようにすることが重要です。

OpenShift Virtualization VM の永続ディスクは、OpenShift Cluster に統合されたONTAPストレージによってバックアップできます。["TridentCSI"](#)。このセクションでは["Tridentプロテクト"](#)データボリュームを含む VM のスナップショットとバックアップをONTAPオブジェクトストレージに作成します。

必要に応じてスナップショットまたはバックアップから復元します。

Trident Protect は、OpenShift クラスター上のアプリケーションと VM のスナップショット、バックアップ、復元、および災害復旧を可能にします。OpenShift Virtualization VM の場合、Trident Protect で保護できるデータには、VM に関連付けられた Kubernetes リソースオブジェクト、永続ボリューム、内部イメージが含まれます。

以下は、このセクションの例で使用されているさまざまなコンポーネントのバージョンです

- "OpenShift クラスター 4.17"
- "Red Hat が提供する OpenShift Virtualization Operator 経由でインストールされた OpenShift Virtualization"
- "Trident25.02"
- "Tridentプロジェクト 25.02"
- "ONTAP 9.16"

オブジェクトストレージ用の**App Vault**を作成する

## AppVaultを作成する

アプリケーションまたはVMのスナップショットとバックアップを作成する前に、スナップショットとバックアップを保存するためにTrident Protectでオブジェクトストレージを構成する必要があります。これはバケットCRを使用して行われます。バケットCRを作成して構成できるのは管理者のみです。バケットCRは、Trident ProtectではAppVaultと呼ばれます。AppVaultオブジェクトは、ストレージバケットの宣言型Kubernetesワークフロー表現です。AppVault CRには、バックアップ、スナップショット、復元操作、SnapMirrorレプリケーションなどの保護操作でバケットを使用するために必要な構成が含まれています。

この例では、ONTAP S3をオブジェクトストレージとして使用する方法を示します。ONTAP S3用のAppVault CRを作成するワークフローは次のとおりです。1. ONTAPクラスタのSVMにS3オブジェクトストアサーバを作成します。2. オブジェクトストアサーバーにバケットを作成します。3. SVMにS3ユーザーを作成します。アクセスキーとシークレットキーを安全な場所に保管してください。4. OpenShiftで、ONTAP S3認証情報を保存するシークレットを作成します。5. ONTAP S3用のAppVaultオブジェクトを作成する

- ONTAP S3用にTrident Protect AppVaultを構成する\*\*

```
# alias tp='tridentctl-protect'

# cat appvault-secret.yaml
apiVersion: v1
stringData:
  accessKeyID: "<access key of S3>"
  secretAccessKey: "<secret access key of S3>"
# you can also provide base 64 encoded values instead of string values
#data:
# base 64 encoded values
#  accessKeyID: < base 64 encoded access key>
#  secretAccessKey: <base 64 encoded secretAccess key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque

# cat appvault.yaml
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
spec:
  providerConfig:
    azure:
      accountName: ""
      bucketName: ""
```

```

    endpoint: ""
gcp:
  bucketName: ""
  projectID: ""
s3:
  bucketName: trident-protect
  endpoint: <lif for S3 access>
  secure: "false"
  skipCertValidation: "true"
providerCredentials:
  accessKeyID:
    valueFromSecret:
      key: accessKeyID
      name: appvault-secret
  secretAccessKey:
    valueFromSecret:
      key: secretAccessKey
      name: appvault-secret
providerType: Ontaps3

# oc create -f appvault-secret.yaml -n trident-protect
# oc create -f appvault.yaml -n trident-protect

```

[root@localhost VM-DataProtection]# tp get appvault						
NAME	PROVIDER	STATE	ERROR	MESSAGE	AGE	
ontap-s3-appvault	OntapS3	Available			8d17h	

## OpenShift Virtualization で VM を作成する

## OpenShift Virtualization で VM を作成する

次のスクリーンショットは、テンプレートを使用してコンソールから VM (名前空間 demo 内の demo-fedora) を作成する様子を示しています。ルートディスクはデフォルトのストレージクラスを自動的に選択するため、デフォルトのストレージクラスが適切に設定されていることを確認します。このセットアップでは、デフォルトのストレージクラスは **sc-zonea-san** です。追加のディスクを作成するときは、ストレージクラス sc-zonea-san を選択し、「最適化されたストレージ設定を適用する」チェックボックスをオンにしてください。これにより、アクセスモードが RWX に、ボリュームモードがブロックに設定されます。



Trident は、SAN (iSCSI、NVMe/TCP、FC) のブロックボリュームモードで RWX アクセスモードをサポートします。(NAS のデフォルトのアクセスモードです)。後で VM のライブマイグレーションを実行する必要がある場合は、RWX アクセスモードが必要です。

### StorageClasses

Name	Provisioner
SC sc-zonea-san - Default	csi.trident.netapp.io
SC thin-csi	csi.vsphere.vmware.com

### VirtualMachines

Catalog

- VirtualMachines
- Templates
- InstanceTypes
- Preferences

Bootable volumes

MigrationPolicies

Checkups

Migration

Networking

Storage

PersistentVolumes

Project: demo

No VirtualMachines found

Click Create VirtualMachine to create your first VirtualMachine or view the catalog tab to...

Create VirtualMachine

- From InstanceType
- From template
- With YAML

**Fedora VM**

fedora-server-small

**Template info**

**Operating system**  
Fedora VM

**Workload type**  
Server (default)

**Description**  
Template for Fedora Linux 39 VM or newer. A PVC with the Fedora disk image must be

**Storage**

Boot from CD ?

**Disk source \*** ?

Template default

**Disk size \***

- 30 + GiB

**Quick create VirtualMachine**

**VirtualMachine name \*** demo-fedora

**Project** demo   **Public SSH key** Not configured edit

Start this VirtualMachine after creation

**Activate Windows** Go to Settings to activate Windows.

**Quick create VirtualMachine** **Customize VirtualMachine**

**Catalog**

**Customize and create VirtualMachine** YAML

Template: Fedora VM

Overview   YAML   Scheduling   Environment   Network interfaces   **Disks**   Scripts   Metadata

**Add disk** ▼

Empty disk (blank)  
Create a disk with no contents.

Use existing

Ephemeral disk (Container image)  
Any changes are lost upon reboot.

Volume  
Add a volume already available on the cluster.

Volume snapshot  
Add a snapshot available on the cluster to the VirtualMachine.

Clone volume  
Clone a volume available on the cluster and add it to the VirtualMachine.

Mount Windows drivers disk

Drive	Interface	Storage class
Disk	virtio	-
Disk	virtio	-

**Create VirtualMachine** **Cancel**

**Activate Windows** Go to Settings to activate Windows.

## Add disk

disk1



Disk size \*

- 30 + GiB ▾

Type

Disk



Hot plug is enabled only for "Disk" and "Lun" types

Interface \*

VirtIO



Hot plug is enabled only for "SCSI" interface

StorageClass

SC SC-zonea-san



Apply optimized StorageProfile settings

Optimized values Access mode: ReadWriteMany, Volume mode: Block.



Save

Cancel

Project: demo

Overview YAML Scheduling Environment Network interfaces Disks Scripts Metadata

Add disk ▾

Filter Search by name...  Mount Windows drivers disk

Name	Source	Size	Drive	Interface	Storage class
cloudinitdisk	Other	-	Disk	virtio	-
disk1	Other	30 GiB	Disk	virtio	sc-zonea-san
rootdisk	PVC (auto import)	30 GiB	Disk	virtio	-
	bootable				

Start this VirtualMachine after creation

Activate Windows  
Go to Settings to activate V

Create VirtualMachine Cancel

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME                                AGE      STATUS    READY
virtualmachine.kubevirt.io/demo-fedora 2m54s   Running   True

NAME          READY   STATUS    RESTARTS   AGE
pod/virt-launcher-demo-fedora-69cts  1/1     Running   0          110s

NAME                           STATUS    VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS
persistentvolumeclaim/demo-fedora  Bound    pvc-406d9d57-4a5e-4123-93c1-13070b7b0140  30Gi     RWO         sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound    pvc-311f2f81-1d25-4a9a-b0cb-836b4e702f04  30Gi     RWO         sc-zonea-san
[root@localhost VM-DataProtection]#
```

アプリを作成する

## アプリを作成

### VM 用の **Trident Protect** アプリを作成する

この例では、デモ名前空間には 1 つの VM があり、アプリの作成時に名前空間のすべてのリソースが含まれます。

```
# alias tp='tridentctl-protect'
# tp create app demo-vm --namespaces demo -n demo --dry-run > app.yaml

# cat app.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: demo-vm
  namespace: demo
spec:
  includedNamespaces:
  - namespace: demo
# oc create -f app.yaml -n demo
```

```
[root@localhost VM-DataProtection]# tp get app -n demo
+-----+-----+-----+
| NAME   | NAMESPACES | STATE | AGE  |
+-----+-----+-----+
| demo-vm | demo      | Ready | 45s |
+-----+-----+-----+
[root@localhost VM-DataProtection]# ■
```

## バックアップを作成してアプリを保護する

## バックアップを作成する

### オンデマンド バックアップを作成する

以前に作成したアプリ (demo-vm) のバックアップを作成します。これには、デモ名前空間内のすべてのリソースが含まれます。バックアップを保存する AppVault 名を指定します。

```
# tp create backup demo-vm-backup-on-demand --app demo-vm --appvault
ontap-s3-appvault -n demo
Backup "demo-vm-backup-on-demand" created.
```

```
[root@localhost VM-DataProtection]# tp get backup -n demo
+-----+-----+-----+-----+-----+
| NAME | APP | RECLAIM POLICY | STATE | ERROR | AGE |
+-----+-----+-----+-----+-----+
| demo-vm-backup-on-demand | demo-vm | Retain | Completed | | 12m53s |
+-----+-----+-----+-----+-----+
[root@localhost VM-DataProtection]# ■
```

### スケジュールに従ってバックアップを作成する

保持するバックアップの粒度と数を指定して、バックアップのスケジュールを作成します。

```

# tp create schedule backup-schedule1 --app demo-vm --appvault ontap-s3-appvault --granularity Hourly --minute 45 --backup-retention 1 -n demo --dry-run>backup-schedule-demo-vm.yaml
schedule.protect.trident.netapp.io/backup-schedule1 created

#cat backup-schedule-demo-vm.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: backup-schedule1
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  backupRetention: "1"
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  hour: ""
  minute: "45"
  recurrenceRule: ""
  snapshotRetention: "0"
status: {}
# oc create -f backup-schedule-demo-vm.yaml -n demo

```

```
[root@localhost VM-DataProtection]# tp get schedule -n demo
```

NAME	APP	SCHEDULE	ENABLED	STATE	ERROR	AGE
backup-schedule1	demo-vm	Hourly:min=45	true			9s

```
[root@localhost VM-DataProtection]# tp get backups -n demo
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
demo-vm-backup-on-demand	demo-vm	Retain	Completed		44m4s
hourly-4c094-20250312144500	demo-vm	Retain	Completed		20m34s

## バックアップからの復元

## バックアップからの復元

### VMを同じ名前空間に復元する

この例では、バックアップ demo-vm-backup-on-demand に、fedora VM の demo-app を含むバックアップが含まれています。

まず、VMを削除し、PVC、ポッド、VMオブジェクトが名前空間「demo」から削除されていることを確認します。

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME                                AGE      STATUS     READY
virtualmachine.kubevirt.io/demo-fedora   59m    Running   True

NAME          READY  STATUS    RESTARTS  AGE
pod/virt-launcher-demo-fedora-69cts  1/1    Running   0          58m

NAME          STATUS    VOLUME          CAPACITY
CLASS  VOLUME ATTRIBUTES CLASS          AGE
persistentvolumeclaim/demo-fedora      Bound   pvc-406d9d57-4a5e-4123-93c1-13070b7b0140  30Gi
a-san <unset>           59m
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-311f2f81-1d25-4a9a-b0cb-836b4e702f04  30Gi
a-san <unset>           59m
[root@localhost VM-DataProtection]#
[root@localhost VM-DataProtection]# oc delete vm demo-fedora -n demo
virtualmachine.kubevirt.io "demo-fedora" deleted
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
No resources found in demo namespace.
[root@localhost VM-DataProtection]# ■
```

Activate  
Go to Settings

次に、バックアップインプレース復元オブジェクトを作成します。

```

# tp create bir demo-fedora-restore --backup demo/demo-vm-backup-on-demand -n demo --dry-run>vm-demo-bir.yaml

# cat vm-demo-bir.yaml
apiVersion: protect.trident.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  annotations:
    protect.trident.netapp.io/max-parallel-restore-jobs: "25"
  creationTimestamp: null
  name: demo-fedora-restore
  namespace: demo
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-0a7b3d353e13/backups/demo-vm-backup-on-demand_f6af3513-9739-480e-88c7-4cca45808a80
  appVaultRef: ontap-s3-appvault
  resourceFilter: {}
status:
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo-bir.yaml -n demo
backupinplacorestore.protect.trident.netapp.io/demo-fedora-restore
created

```

```

[root@localhost VM-DataProtection]# tp get bir -n demo
+-----+-----+-----+-----+
|     NAME      | APPVAULT      | STATE      | ERROR | AGE      |
+-----+-----+-----+-----+
| demo-fedora-restore | ontap-s3-appvault | Completed |       | 28m17s |
+-----+-----+-----+-----+
[root@localhost VM-DataProtection]#

```

VM、ポッド、PVCが復元されたことを確認する

```

[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME                                     AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-fedora   116s  Running  True

NAME          READY  STATUS    RESTARTS  AGE
pod/virt-launcher-demo-fedora-9kfxh     1/1    Running   0        116s

NAME                           STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS
persistentvolumeclaim/demo-fedora      Bound   pvc-6f69a62c-285c-4980-b0dd-6c85baccf346  30Gi     RWX          sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8  30Gi     RWX          sc-zonea-san
[root@localhost VM-DataProtection]#

```

VM を別の名前空間に復元する

まず、アプリを復元する新しい名前空間（この例では demo2）を作成します。次に、バックアップ復元オブジェクトを作成します

```
# tp create br demo2-fedora-restore --backup demo/hourly-4c094-20250312154500 --namespace-mapping demo:demo2 -n demo2 --dry-run>vm-demo2-br.yaml

# cat vm-demo2-br.yaml
apiVersion: protect.trident.netapp.io/v1
kind: BackupRestore
metadata:
  annotations:
    protect.trident.netapp.io/max-parallel-restore-jobs: "25"
  creationTimestamp: null
  name: demo2-fedora-restore
  namespace: demo2
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-0a7b3d353e13/backups/hourly-4c094-20250312154500_aaa14543-a3fa-41f1-a04c-44b1664d0f81
  appVaultRef: ontap-s3-appvault
  namespaceMapping:
    - destination: demo2
      source: demo
  resourceFilter: {}
status:
  conditions: null
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo2-br.yaml -n demo2
```

[root@localhost VM-DataProtection]# tp get br -n demo2					
NAME	APPVAULT	STATE	ERROR	AGE	
demo2-fedora-restore	ontap-s3-appvault	Completed		38m52s	

VM、ポッド、および PVC が新しい名前空間 demo2 に作成されていることを確認します。

```
[root@localhost VM-DataProtection]#  
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2  
NAME AGE STATUS READY  
virtualmachine.kubevirt.io/demo-fedora 5m8s Running True  
  
NAME READY STATUS RESTARTS AGE  
pod/virt-launcher-demo-fedora-c7xc6 1/1 Running 0 5m7s  
  
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS  
persistentvolumeclaim/demo-fedora Bound pvc-4d278ae2-76cc-46f1-bbf8-071ae75e4a82 30Gi RWX sc-zonea-san  
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87 Bound pvc-9b96d09c-7226-4ffc-829b-2cee88e7a117 30Gi RWX sc-zonea-san  
[root@localhost VM-DataProtection]#
```

## スナップショットを使用してアプリを保護する

## スナップショットを作成する

オンデマンド スナップショットを作成する アプリのスナップショットを作成し、保存する必要がある AppVault を指定します。

```
# tp create snapshot demo-vm-snapshot-on-demand --app demo-vm --appvault
ontap-s3-appvault -n demo --dry-run
# cat demo-vm-snapshot-on-demand.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  creationTimestamp: null
  name: demo-vm-snapshot-on-demand
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  completionTimeout: 0s
  volumeSnapshotsCreatedTimeout: 0s
  volumeSnapshotsReadyToUseTimeout: 0s
status:
  conditions: null
  postSnapshotExecHooksRunResults: null
  preSnapshotExecHooksRunResults: null
  state: ""

# oc create -f demo-vm-snapshot-on-demand.yaml
snapshot.protect.trident.netapp.io/demo-vm-snapshot-on-demand created
```

```
[root@localhost VM-DataProtection]#
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2
NAME           AGE   STATUS    READY
virtualmachine.kubevirt.io/demo-fedora   5m8s   Running   True

NAME          READY   STATUS    RESTARTS   AGE
pod/virt-launcher-demo-fedora-c7xc6   1/1     Running   0          5m7s

NAME                           STATUS   VOLUME
persistentvolumeclaim/demo-fedora   Bound    pvc-4d278ae2-76cc-46f1-bbf8-071ae75e4a82   30Gi   RWX   sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87   Bound    pvc-9b96d09c-7226-4ffc-829b-2cee88e7a117   30Gi   RWX   sc-zonea-san
[root@localhost VM-DataProtection]#
```

スナップショットのスケジュールを作成する スナップショットのスケジュールを作成します。保持するスナップショットの粒度と数を指定します。

```

# tp create Schedule snapshot-schedule1 --app demo-vm --appvault ontap-s3-appvault --granularity Hourly --minute 50 --snapshot-retention 1 -n demo --dry-run>snapshot-schedule-demo-vm.yaml

# cat snapshot-schedule-demo-vm.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: snapshot-schedule1
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  backupRetention: "0"
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  hour: ""
  minute: "50"
  recurrenceRule: ""
  snapshotRetention: "1"
status: {}

# oc create -f snapshot-schedule-demo-vm.yaml
schedule.protect.trident.netapp.io/snapshot-schedule1 created

```

```
[root@localhost VM-DataProtection]# tp get schedule -n demo
```

NAME	APP	SCHEDULE	ENABLED	STATE	ERROR	AGE
backup-schedule1	demo-vm	Hourly:min=45	true			5d23h
snapshot-schedule1	demo-vm	Hourly:min=50	true			12s

```
[root@localhost VM-DataProtection]#
```

```
[root@localhost VM-DataProtection]# tp get snapshots -n demo
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
backup-39b67e1c-f875-4045-93df-78634bae9dfb	demo-vm	Delete	Completed		6m29s
demo-vm-snapshot-on-demand	demo-vm	Delete	Completed		21m30s
hourly-51839-20250318135000	demo-vm	Delete	Completed		1m29s

```
[root@localhost VM-DataProtection]#
```

## スナップショットからの復元

## スナップショットからの復元

スナップショットから同じ名前空間に VM を復元します demo2 名前空間から VM demo-fedora を削除します。

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                                     AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-fedora    28h   Running  True

NAME                                     STATUS  VOLUME
DRAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-fedora          Bound   pvc-e0d5f79d-dff9-450d-be0e-90ab6880b7af
-zonea-san <unset>                  28h
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-e6f7890a-70c7-4538-9035-5e2e9379511d
-zonea-san <unset>                  28h
[root@localhost RedHat]# oc delete virtualmachine.kubevirt.io/demo-fedora -n demo
virtualmachine.kubevirt.io "demo-fedora" deleted
[root@localhost RedHat]# ■
```

Act  
Go t

VM のスナップショットからスナップショットインプレース復元オブジェクトを作成します。

```
# tp create sir demo-fedora-restore-from-snapshot --snapshot demo/demo-
vm-snapshot-on-demand -n demo --dry-run>vm-demo-sir.yaml

# cat vm-demo-sir.yaml
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  creationTimestamp: null
  name: demo-fedora-restore-from-snapshot
  namespace: demo
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-
0a7b3d353e13/snapshots/20250318132959_demo-vm-snapshot-
on-demand_e3025972-30c0-4940-828a-47c276d7b034
  appVaultRef: ontap-s3-appvault
  resourceFilter: {}
status:
  conditions: null
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo-sir.yaml
snapshotinplacelrestore.protect.trident.netapp.io/demo-fedora-restore-
from-snapshot created
```

```
[root@localhost VM-DataProtection]# tp get sir -n demo
```

NAME	APPVAULT	STATE	ERROR	AGE
demo-fedora-restore-from-snapshot	ontap-s3-appvault	Completed		58m17s

```
[root@localhost VM-DataProtection]#
```

VM とその PVC がデモ名前空間に作成されていることを確認します。

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                                AGE      STATUS    READY
virtualmachine.kubevirt.io/demo-fedora 5m17s   Running   True

NAME                                     STATUS    VOLUME
persistentvolumeclaim/demo-fedora        Bound    pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound    pvc-db085154-079f-45ad-9e62-9656e913d01c
[root@localhost RedHat]#
[root@localhost RedHat]#
```

スナップショットから別の名前空間にVMを復元する

以前にバックアップから復元した demo2 名前空間内の VM を削除します。

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2
NAME                                AGE      STATUS    READY
virtualmachine.kubevirt.io/demo-centos 3m12s   Running   True
virtualmachine.kubevirt.io/demo-fedora  3m11s   Running   True

NAME          READY  STATUS    RESTARTS  AGE
pod/virt-launcher-demo-centos-w77rr  1/1    Running   0          3m11s
pod/virt-launcher-demo-fedora-wwdtc  1/1    Running   0          3m11s

NAME           STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-centos    sc-zonea-san <unset>            3m26s
persistentvolumeclaim/demo-fedora   sc-zonea-san <unset>            3m25s
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 sc-zonea-san <unset>            3m20s
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87    sc-zonea-san <unset>            3m16s
[root@localhost VM-DataProtection]# oc delete virtualmachine.kubevirt.io/demo-fedora -n demo2
virtualmachine.kubevirt.io "demo-fedora" deleted
```

スナップショットからスナップショット復元オブジェクトを作成し、名前空間マッピングを提供します。

```

# tp create sr demo2-fedora-restore-from-snapshot --snapshot demo/demo-
vm-snapshot-on-demand --namespace-mapping demo:demo2 -n demo2 --dry
-run>vm-demo2-sr.yaml

# cat vm-demo2-sr.yaml
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotRestore
metadata:
  creationTimestamp: null
  name: demo2-fedora-restore-from-snapshot
  namespace: demo2
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-
0a7b3d353e13/snapshots/20250318132959_demo-vm-snapshot-
on-demand_e3025972-30c0-4940-828a-47c276d7b034
  appVaultRef: ontap-s3-appvault
  namespaceMapping:
    - destination: demo2
      source: demo
  resourceFilter: {}
status:
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo2-sr.yaml
snapshotrestore.protect.trident.netapp.io/demo2-fedora-restore-from-
snapshot created

```

[root@localhost VM-DataProtection]# tp get sr -n demo2				
NAME	APPVAULT	STATE	ERROR	AGE
demo2-fedora-restore-from-snapshot	ontap-s3-appvault	Completed		15m22s

VM とその PVC が新しい名前空間 demo2 に復元されていることを確認します。

[root@localhost RedHat]# oc get vm,pvc -n demo2				
NAME	AGE	STATUS	READY	
virtualmachine.kubevirt.io/demo-fedora	29h	Running	True	
NAME	AGE	STATUS	VOLUME	
DRAGECLASS VOLUMEATTRIBUTESCLASS				
persistentvolumeclaim/demo-fedora				
-zonea-san <unset>	29h	Bound	pvc-35dc9b2-4fca-486c-af9e-596bc5b4dc15	Ac Go
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87				
-zonea-san <unset>	29h	Bound	pvc-575a3111-382f-4933-a778-0089fa1ea2af	Ac Go

特定の**VM**を復元する

名前空間内の特定の**VM**を選択してスナップショット/バックアップを作成し、復元する

前の例では、名前空間内に 1 つの VM がありました。名前空間全体をバックアップに含めることで、その VM に関連付けられたすべてのリソースがキャプチャされました。次の例では、同じ名前空間に別の VM を追加し、ラベルセレクターを使用してこの新しい VM 専用のアプリを作成します。

デモ名前空間に新しいVM（demo-centos vm）を作成します

```
[root@localhost VM-DataProtection]# oc get vm,pod,pvc -n demo
NAME                                AGE     STATUS    READY
virtualmachine.kubevirt.io/demo-centos 2m47s  Running  True
virtualmachine.kubevirt.io/demo-fedora  81m    Running  True

NAME          READY  STATUS    RESTARTS  AGE
pod/virt-launcher-demo-centos-2nq6g  1/1    Running  0          113s
pod/virt-launcher-demo-fedora-9kfxh  1/1    Running  0          81m

NAME                           STATUS    VOLUME
persistentvolumeclaim/demo-centos  Bound    pvc-ed0f492b-0109-471d-b395-9077ae5f1fa7
persistentvolumeclaim/demo-fedora  Bound    pvc-6f69a62c-285c-4980-b0dd-6c85baccf346
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 Bound    pvc-3c01142a-4344-4293-ae67-7d3925c56211
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87   Bound    pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8
CAPACITY  ACCESS MODES  STORAGECLASS
30Gi      RWX         sc-zonea-san
30Gi      RWX         sc-zonea-san
30Gi      RWX         sc-zonea-san
30Gi      RWX         sc-zonea-san
[root@localhost VM-DataProtection]# _
```

**demo-centos vm** とその関連リソースにラベルを付ける

```
[root@localhost VM-DataProtection]# oc label vm demo-centos category=protect-demo-centos -n demo  
virtualmachine.kubevirt.io/demo-centos labeled  
[root@localhost VM-DataProtection]# oc label pvc demo-centos category=protect-demo-centos -n demo  
persistentvolumeclaim/demo-centos labeled  
[root@localhost VM-DataProtection]# oc label pvc dv-demo-centos-lavender-tortoise-34 category=protect-demo-centos -n demo  
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 labeled  
[root@localhost VM-DataProtection]#
```

**demo-centos vm** と **pvcs** にラベルが付与されていることを確認します

```
[root@localhost VM-DataProtection]# oc get vm --show-labels -n demo
NAME     AGE   STATUS  LABELS
demo-centos 6m33s  Running  True    app=demo-centos,category=protect-demo-centos,kubevirt.io/dynamic-credentials-support=true,vm.kubevirt.io/template.namespace=openshift,vm.kubevirt.io/template.revision=1,vm.kubevirt.io/template.version=v0.31.1,vm.kubevirt.io/template=centos-stream9-server-small
demo-fedora  85m  Running  True    app=demo-fedora,vm.kubevirt.io/template.namespace=openshift,vm.kubevirt.io/template.revision=1,vm.kubevirt.io/template.version=v0.31.1,vm.kubevirt.io/template=fedora-server-small
[root@localhost VM-DataProtection]#
```

```
[root@localhost VM-DataProtection]# oc get pvc --show-labels -n demo
NAME           STATUS  VOLUME                                     CAPACITY   ACCESS MODES  STORAGECLASS          VOLUMEHANDLE  ATTRIBUTESCLASS  AGE    LABELS
demo-centos   Bound   pvc-adf492b-0109-471d-b5c5-9877ae5f1a7  30Gi      RWM          sc-zone-a-sam  cunset        7m41s  app.kubernetes.io/component=storage,app.kubernetes.io/instance=demo-centos,instancetype=kubevirt.io/default-instance-type,u.medium,instancetype=u.medium,instance-type=kubevirt.io/default-preference=centos,stream=kubevirt.io:created-by=a6a7b049-669a-4e21-a78-20743671b284
demo-fedora   Bound   pvc-f6f969a2c-285c-4980-b04d-0c850accf306  30Gi      RWM          sc-zone-a-sam  <unset>       86m   app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hypervisor-converged-cluster,app.kubernetes.io/version=4.17.5,app-containerized-data-importer,category=protect-demo-centos,instancetype=kubevirt.io/default-instance-type,u.default-preference=fedora,kubevirt.io:created-by=7d5184e9-22f8-4456-9afe-3d1984c3d099
dv_demo-centos-lambda-tortoise-34  Bound   pvc-3c0112a4-4344-4293-a677-7d39d526111  30Gi      RWK          sc-zone-a-sam  <unset>       7m41s  app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hypervisor-converged-cluster,app.kubernetes.io/version=4.17.5,app-containerized-data-importer,category=protect-demo-centos,kubevirt.io:created-by=7d5184e9-22f8-4456-9afe-3d1984c3d099
dv_demo-fedora-fushcia-shreu-87  Bound   pvc-81bd2828-7aca-40fc-8f8f-6e99246e6f38  30Gi      RWX          sc-zone-a-sam  <unset>       86m   app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hypervisor-converged-cluster,app.kubernetes.io/version=4.17.5,app-containerized-data-importer,kubevirt.io:created-by=7d5184e9-22f8-4456-9afe-3d1984c3d099
[root@localhost VM-DataProtection]#
```

ラベルセレクターを使用して特定のVM (`demo-centos`) 専用のアプリを作成する

```

# tp create app demo-centos-app --namespaces 'demo(category=protect-
demo-centos)' -n demo --dry-run>demo-centos-app.yaml

# cat demo-centos-app.yaml

apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: demo-centos-app
  namespace: demo
spec:
  includedNamespaces:
  - labelSelector:
    matchLabels:
      category: protect-demo-centos
      namespace: demo
status:
  conditions: null

# oc create -f demo-centos-app.yaml -n demo
application.protect.trident.netapp.io/demo-centos-app created

```

[root@localhost VM-DataProtection]# tp get app -n demo				
NAME	NAMESPACES	STATE	AGE	
demo-centos-app	demo	Ready	56s	
demo-vm	demo	Ready	4h6m	

オンデマンドおよびスケジュールに従ってバックアップとスナップショットを作成する方法は、前に示したものと同じです。スナップショットまたはバックアップの作成に使用されている trident-protect アプリには名前空間の特定の VM のみが含まれているため、そこから復元すると特定の VM のみが復元されます。以下に、サンプルのバックアップ/復元操作の例を示します。

対応するアプリを使用して、名前空間内の特定の VM のバックアップを作成します

前の手順では、ラベル セレクターを使用して、デモ名前空間に centos vm のみを含むアプリを作成しました。このアプリのバックアップ(この例ではオンデマンド バックアップ)を作成します。

```
# tp create backup demo-centos-backup-on-demand --app demo-centos-app
--appvault ontap-s3-appvault -n demo
Backup "demo-centos-backup-on-demand" created.
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
demo-centos-backup-on-demand	demo-centos-app	Retain	Completed		13m22s
demo-vm-backup-on-demand	demo-vm	Retain	Completed		4h19m
hourly-4c094-20250312174500	demo-vm	Retain	Completed		56m17s

特定の VM を同じ名前空間に復元する 特定の VM (centos) のバックアップは、対応するアプリを使用して作成されました。これからバックアップ・インプレース・リストアまたはバックアップ・リストアが作成されると、この特定の VM のみが復元されます。Centos VM を削除します。

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                           AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-centos  4m27s  Running True
virtualmachine.kubevirt.io/demo-fedora  4m27s  Running True

NAME                                     STATUS  VOLUME
STORAGECLASS    VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-centos        Bound   pvc-e8faeaf8-fc0c-4d92-96de-c83a335a7a17
  sc-zonea-san <unset>                4m33s
persistentvolumeclaim/demo-fedora        Bound   pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
  sc-zonea-san <unset>                4m33s
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34  Bound   pvc-66eb7996-1420-4513-a67c-2824f08534da
  sc-zonea-san <unset>                4m33s
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87     Bound   pvc-db085154-079f-45ad-9e62-9656e913d01c
  sc-zonea-san <unset>                4m32s
```

```
[root@localhost RedHat]# oc delete virtualmachine.kubevirt.io/demo-centos -n demo
virtualmachine.kubevirt.io "demo-centos" deleted
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                           AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-fedora  5m17s  Running True

NAME                                     STATUS  VOLUME
persistentvolumeclaim/demo-fedora        Bound   pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-db085154-079f-45ad-9e62-9656e913d01c
[root@localhost RedHat]#
```

demo-centos-backup-on-demand からバックアップのインプレース リストアを作成し、centos VM が再作成されたことを確認します。

```
#tp create bir demo-centos-restore --backup demo/demo-centos-backup-on-
demand -n demo
BackupInplaceRestore "demo-centos-restore" created.
```

```
[root@localhost RedHat]# tp get bir -n demo
```

NAME	APPVAULT	STATE	ERROR	AGE
demo-centos-restore	ontap-s3-appvault	Completed		57m9s
demo-fedora-restore	ontap-s3-appvault	Completed		7d5h

```
[root@localhost RedHat]# oc get vm,pvc -n demo
```

NAME	AGE	STATUS	READY
virtualmachine.kubevirt.io/demo-centos	29m	Running	True
virtualmachine.kubevirt.io/demo-fedora	85m	Running	True

NAME	STORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE	STATUS	VOLUME
persistentvolumeclaim/demo-centos	sc-zonea-san	<unset>	29m	Bound	pvc-82954bf7-4a7e-4e0c-9a04-4fa152e1b0ef
persistentvolumeclaim/demo-fedora	sc-zonea-san	<unset>	85m	Bound	pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34	sc-zonea-san	<unset>	29m	Bound	pvc-2a8d4eb5-ed6d-4408-b85d-e218e9a5d4b0
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87	sc-zonea-san	<unset>	85m	Bound	pvc-db085154-079f-45ad-9e62-9656e913d01c

```
[root@localhost RedHat]#
```

特定の VM を別の名前空間に復元する demo-centos-backup-on-demand から別の名前空間 (demo3) へのバックアップ復元を作成し、centos VM が再作成されたことを確認します。

```
# tp create br demo2-centos-restore --backup demo/demo-centos-backup-on-demand --namespace-mapping demo:demo3 -n demo3
BackupRestore "demo2-centos-restore" created.
```

```
[root@localhost RedHat]#
```

```
[root@localhost RedHat]# tp get br -n demo3
```

NAME	APPVAULT	STATE	ERROR	AGE
demo2-centos-restore	ontap-s3-appvault	Completed		52m57s

```
[root@localhost RedHat]#
```

```
[root@localhost RedHat]# oc get vm,pvc -n demo3
```

NAME	AGE	STATUS	READY
virtualmachine.kubevirt.io/demo-centos	19m	Running	True

NAME	STORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE	STATUS	VOLUME
persistentvolumeclaim/demo-centos	sc-zonea-san	<unset>	19m	Bound	pvc-0a14e38f-07de-4e09-8f88-14a9a8bb45c2
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34	sc-zonea-san	<unset>	19m	Bound	pvc-d4f9cf2f-264c-4d02-94bf-0db28b427acc

## ビデオモンストレーション

次のビデオは、スナップショットを使用してVMを保護するデモを示しています。

[VMの保護](#)

## 著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を隨時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5225.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。